A Closer Look at Selective DFT Attacks

Guang Gong
Department of Electrical and Computer Engineering
University of Waterloo, Waterloo, Ontario, Canada
Email: ggong@uwaterloo.ca

Abstract

In this essay, we revisit the selective discrete Fourier transform (DFT) attacks. We first demonstrate some comparisons of the time domain and frequency domain properties of the DFT of sequences over a finite field with those of DFT of real or complex number functions. Following this, a fast computation of DFT is given using the selective DFT algorithm. Finally, we introduce reference pairs to simplify the selective DFT attacks and their applications to attack on hash functions and block ciphers.

1 Introduction

Recently, recovering a complete internal state in a stream cipher generator by making use of the discrete Fourier transform of sequences is introduced by the research teams in Waterloo and Bergen. This idea is originally presented in [12] without using the DFT for some special case. Later on, it is generalized to a more general case by introducing the selective DFT approach in [13] [4].

Selective discrete Fourier transform attacks are inspired by modulation techniques in communications. For a cosine waveform $g(x) = \cos(2\pi f_c t)$ in the doubly infinite interval, it is continues in the time domain and have infinite many nonzero values. However, its Fourier transform, defined as

$$G(f) = \int_{-\infty}^{\infty} \cos(2\pi f_c t) e^{-2\pi f j t} dt \tag{1}$$

where $j = \sqrt{-1}$, is given by

$$G(f) = \frac{1}{2}[\delta(f - f_c) + \delta(f + f_c)]$$

where $\delta(x)$ is defined by

$$\delta(x) = \begin{cases} 1 & \text{if } x = 0 \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

and f is a variable, called a *frequency variable*. In other words, in the frequency domain, it has only two nonzero values, which are shown in Figure 1.

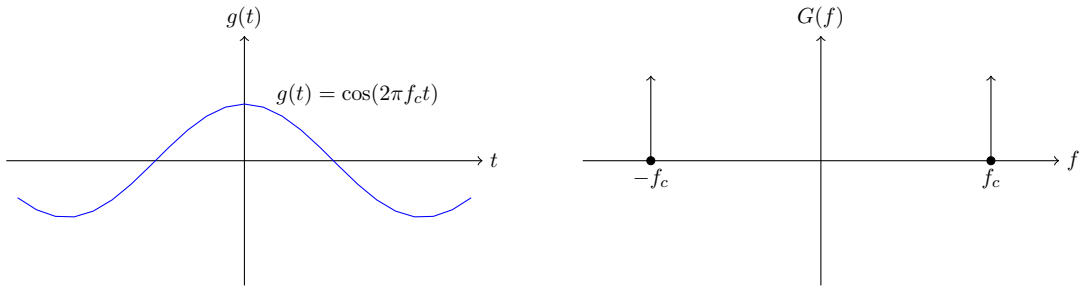

Figure 1: Cosine function and its Fourier spectral density

Due to this fact, in the design of transmission systems, the properties, such as to compute the average power of the cosine function and bandwidth of the cosine signal, determined by an interval of $f > 0$ for which $G(f)$ has the nonzero values, etc., are analyzed in the frequency domain.

In this essay, we attempt to simply the selective DFT attack by introducing reference pairs in the attack. In Section 2, we review the DFT of sequences. In Section 3, we present a fast algorithm for computing the discrete Fourier transform (DFT) without knowing the entire sequence bits in terms of the idea of the selective DFT attack. In Section 4, a simplified selective DFT attack is given using a reference pair in the attack and extend it to attack hash functions and block ciphers. Section 5 concludes the essay.

2 Discrete Fourier Transform of Binary Sequences

Now we consider a binary sequence of length N, say $\mathbf{a} = (a_0, a_1, \cdots, a_{N-}), a_i \in \mathbb{F}_2$. We can also define its Fourier transform. Since $\mathbf{a}$ is discrete, its Fourier transform is called *discrete Fourier transform (DFT)*, defined in terms of a finite field field $GF(2^n)$, denoted as $\mathbb{F}_{2^n}$, where n is the smallest number such that $N \mid 2^n - 1$. In this case, N is odd. Let α be an element in

$\mathbb{F}_{2^n}$ with order N. DFT over a finite field has been throughly treated in many different text books, say [9][2] [8] in different contents.

2.1 Definitions and Basic Properties of DFT

The *Discrete Fourier Transform (DFT)* of $\{a_t\}$ is defined by

$$A_k = \sum_{t=0}^{N-1} a_t \alpha^{-tk}, \ k = 0, 1, \ldots, N-1. \tag{3}$$

The *inverse DFT* , denoted as IDFT, is given by

$$a_t = \sum_{k=0}^{N-1} A_k \alpha^{kt}, \quad t = 0, 1, \ldots, N-1. \tag{4}$$

The sequence $\{A_k\}$ is called a *DFT spectral sequence* of $\mathbf{a}$ (with respect to α) or DFT spectra for short.

Note that the DFT spectral value A_k, in general, is an element in the extension field $\mathbb{F}_{2^n}$, while $\{a_t\}$ is a binary sequence. This is a similar fact as that the Fourier transform of a real valued function could be a complex valued function. For example, for $g(t) = \sin(2\pi f_c t)$, we have $G(f) = \frac{1}{2j}[\delta(f - f_c) + \delta(f + f_c)]$ where $\delta(x)$ is defined by (2), which is a complex function.

Furthermore, since $a_t^2 = a_t$, the uniqueness of the representation of a_t implies the following property.

Property 1 *The DFT spectral sequence satisfies*

$$A_{2k} = A_k^2, \forall k \tag{5}$$

where indices are calculated modulo N.

Thus, for DFT computation, we only need to compute $\{A_k \,|\, k \in I\}$ where I consists of all integers such that $0 \leq k < N$ and $k \not\equiv k' 2^i \pmod{N}$ for any two integers $k, k' \in I$. The integers in I are called *coset leaders* modulo N, which can be computed as follows.

Example 1 Let $\mathbf{a} = (a_0, \cdots, a_6) = 0010111$ with period $N = 7$ (we know that $\mathbf{a}$ is an m-sequence of period 7). Since $N = 7 = 2^3 - 1$, we can generate a finite field $GF(2^3)$, defined by

Table 1: Procedure for computing the index set I

1.	$0 \in I$ and $1 \in I$.
2.	Assume that we have constructed a subset of I, say I_0. Let $k \in I_0$ be the largest number in I_0. We then select $j > k$ such that $j \notin \{k2^i (\text{mod } N)\,] \mid 0 \le i < n\}$ for all $k \in I_0$.
3.	In the same fashion, we obtain I.

Table 2: $GF(2^3)$ defined by $\alpha^3 + \alpha + 1 = 0$ where i is the exponent in α^i, and $\alpha^7 = 1$.

$(\alpha^2, \alpha, 1)$	i	$(\alpha^2, \alpha, 1)$	i
000		011	3
001	0	110	4
010	1	111	5
100	2	101	6

a primitive polynomial $t(x) = x^3+x+1$ where α is a root of $t(x)$ in $GF(2^3)$, i.e., $\alpha^3+\alpha+1 = 0$. Thus $ord(\alpha) = 7$, i.e., $\alpha^7 = 1$ where 7 is the smallest number satisfying that. The elements in $GF(2^3)$ is given in Table 2. According to the definition, the DFT of $\mathbf{a}$ is

$$A_k = \sum_{t=0}^{6} a_t \alpha^{-tk} = \alpha^{-2k} + \alpha^{-4k} + \alpha^{-5k} + \alpha^{-6k}, k = 0, 1, \cdots, 6.$$

We have $I = \{0, 1, 3\}$ modulo 7. Thus we only need to compute A_0, A_1 and A_3. The rest of spectral values can be obtained from (5).

$$\begin{aligned}
A_0 &= \textstyle\sum_{t=0}^{6} a_t = 0, \\
A_1 &= \alpha^{-2} + \alpha^{-4} + \alpha^{-5} + \alpha^{-6} = \alpha^5 + \alpha^3 + \alpha^2 + \alpha \text{ (by } \alpha^7 = 1) = \alpha \text{ (by look-up table)} \\
A_2 &= A_1^2 = \alpha^2, A_4 = A_2^2 = \alpha^4 \qquad \text{(by (5))} \\
A_3 &= \alpha^{-2\times 3} + \alpha^{-4\times 3} + \alpha^{-5\times 3} + \alpha^{-6\times 3} = \alpha + \alpha^2 + \alpha^6 + \alpha^3 = 0 \\
A_5 &= A_3^4 = 0, A_6 = A_3^2 = 0.
\end{aligned}$$

Thus

$$(A_0, A_1, \cdots, A_6) = (0, \alpha, \alpha^2, 0, \alpha^4, 0, 0) = (000, 010, 100, 000, 110, 000, 000)$$

where the last identity comes from writing the elements of $GF(2^3)$ in terms of the vector representation. Both the sequence and its DFT are plotted in Figure 2.

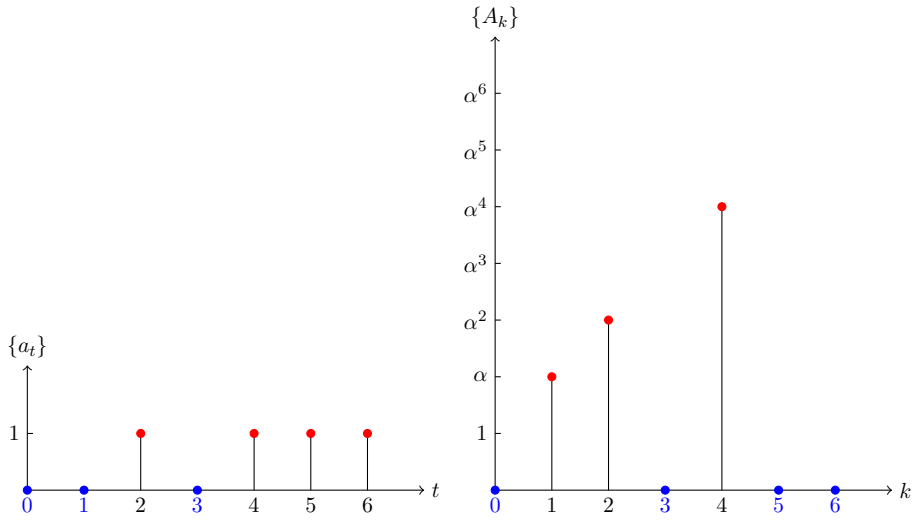

Figure 2: Binary sequence $\{a_t\} = 0010111$ and its DFT $\{A_k\} = (0, \alpha, \alpha^2, 0, \alpha^4, 0, 0)$

Example 2 Let **a** be an m-sequence of degree 4, given by

$$\{a_t\} = 011110101100100.$$

Let $GF(2^4)$ be defined by a primitive polynomial $f(x) = x^4+x+1$ with α is a root of $f(x) = 0$, see Table 3. Using Table 3, we can compute the DFT of **a**:

$$A_k = \sum_{t=0}^{14} a_t \alpha^{-tk}, k = 0, 1, \cdots, 14$$
$$\Longrightarrow A_0 = 0, A_k = 1, k \in \{7, 11, 13, 14\}, A_k = 0 \text{ for the rest of } k.$$

Both the sequence and its DFT are plotted in Figure 3.

We can also write the DFT and IDFT in terms of the polynomials, as shown in the following property.

Table 3: $GF(2^4)$, defined by $\alpha^4 + \alpha + 1 = 0$ where $\underline{\alpha} = (\alpha^3, \alpha^2, \alpha, 1)$, i is the exponent of the corresponding element in α^i and $\alpha^{15} = 1$.

$\underline{\alpha}$	i	$\underline{\alpha}$	i
0000		1011	7
0001	0	0101	8
0100	1	1010	9
0010	2	0111	10
1000	3	1110	11
0011	4	1111	12
0110	5	1101	13
1100	6	1001	14

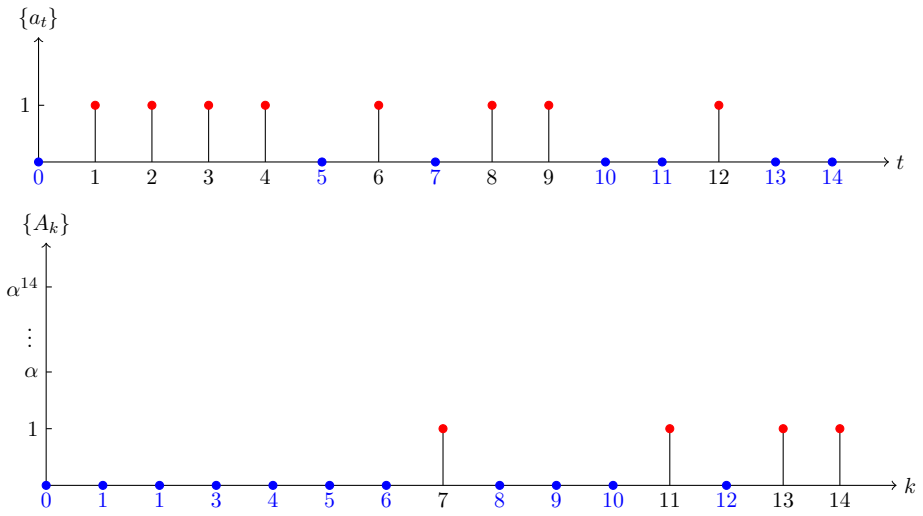

Figure 3: Binary sequence $\{a_t\} = 011110101100100$ and its DFT $\{A_k\} = 000000010001011$

Property 2 *With the above notation, we have*

With $a(x) = \sum_{t=0}^{N-1} a_t x^t$ *and* $A(x) = \sum_{t=0}^{N-1} A_k x^k$, *then*
DFT: $A_k = a(\alpha^{-k}), \quad k = 0, 1, \cdots, N-1$
IDFT: $a_t = A(\alpha^t), \quad t = 0, 1, \cdots, N-1.$

The following property gives the relation between the minimal polynomial and DFT.

Property 3 *Let $h(x)$ be the minimal polynomial which generates the sequence $\{a_t\}$. Then*

$$A_k \neq 0 \Longleftrightarrow h(\alpha^k) = 0, 0 \leq k < N. \tag{6}$$

Furthermore, the linear span of the sequence, the degree of $h(x)$, is equal to the number of nonzero DFT spectral points.

2.2 Some Basic Operations of DFT

The Fourier transform of signals (here we assume that all signals we consider are the functions which satisfy certain conditions in order to have their Fourier transform exists) satisfy a number of interesting properties [11]. In the following, we list a few whose DFT versions will be given here where x^* is the complex conjugate of the complex number x.

Table 4: Some Fourier Transform Properties

Operations	Signal $g(t)$	Fourier Transform $G(f)$
Linearity	$ag(t) + bh(t)$	$aG(f) + bH(f)$
Time delay or time shift	$g(t - \tau)$	$e^{-2\pi f\tau}G(f)$ (phase shift)
Time convolution	$g(t) * h(t)$ $= \int_{-\infty}^{\infty} g(t)h^*(\tau - t)dt$	$G(f)H(f)$
Frequency convolution	$g(t)h(t)$	$G(f) * H(f)$
Parseval formular	$\int_{-\infty}^{\infty} g(t)h^*(t)dt = \int_{-\infty}^{\infty} G(f)H^*(f)dt$	

According to the definition of the DFTs of binary sequences, we know they are linear. In the following, we show the time shift and time convolution of the DFTs.

Property 4 (TIME SHIFT) *Let* $\mathbf{b} = \{b_t\}$ *whose elements are given by* $b_t = a_{t+\tau}$ *and let* $\{B_k\}$ *be its DFT. Then*

$$B_k = \alpha^{k\tau} A_k, k = 0, 1, \cdots, N-1.$$

For example, in Example 1, let $b_t = a_{t+6}, t = 0, \cdots, 6$. Then we have $(b_0, \cdots, b_6) = (1, 0, 0, 1, 0, 1, 1)$ and

$$B_k = \alpha^{6k} A_k, k = 0, \cdots, 6$$

$\Longrightarrow$

$$\begin{aligned} &B_0 = 0, \\ &B_1 = \alpha^6 A_1 = \alpha^6 \alpha = 1 \Longrightarrow B_2 = B_4 = 1, \\ &B_3 = B_5 = B_6 = 0 \text{ (because } A_3 = A_5 = A_6 = 0). \end{aligned}$$

Property 5 *With the same notation, let* $q(x) = \sum_{i=0}^{r} c_i x^i$ *be a polynomial over* $\mathbb{F}_2$ *of degree* r *and* $\{a_t\}$*, and* $\{v_t\}$ *be the discrepancy sequence of applying* $q(x)$ *to* $\{a_t\}$*, defined by*

$$v_t = \sum_{i=0}^{r} c_i a_{i+t}, t = 0, 1, \cdots, N-1. \tag{7}$$

Let $\{V_k\}$ *be the DFT of* $\{v_t\}$*. Then*

$$V_k = q(\alpha^k) A_k, k = 0, 1, \cdots, N-1. \tag{8}$$

Proof 1 *From the IDFT, we have* $a_t = \sum_{k=0}^{N-1} A_k \alpha^{tk}$*, and therefore*

$$v_t = \sum_{i=0}^{r} q_i a_{i+t} = \sum_{i=0}^{r} q_i \sum_{k=0}^{N-1} A_k \alpha^{(i+t)k} = \sum_{k=0}^{N-1} A_k \alpha^{tk} \sum_{i=0}^{r} q_i \alpha^{ik} = \sum_{k=0}^{N-1} A_k q(\alpha^k) \alpha^{tk}.$$

This lemma has two different interpretations.

Case 1. Time convolution. If $r \leq N$, let $\mathbf{q} = \{q_t\}_{t=0}^{N-1}$ where $q_j = 0$ for $j = r+1, \cdots, N-1$, and $\{Q_k\}$ be its DFT. According to Property 2, we have $Q_k = q(\alpha^k)$.

$$\boxed{v_t = \sum_{i=0}^{r} q_i a_{i+t}, t = 0, 1, \cdots, N-1, \quad \Longrightarrow V_k = Q_k A_k, k = 0, 1, \cdots, N-1}$$

Case 2. Passing $\{a_t\}$ *through linear time invariant (LTI) system with function* $q(x)$*.* The generation of $\{v_t\}$ is equivalent to the following linear system, shown in Figure 2.2.

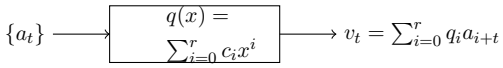

Figure 4: Passing the sequence $\{a_t\}$ through LTI system defined by $q(x)$: $V_k = q(\alpha^k)A_k, k = 0, \cdots, N-1$, and $V_k = Q_k A_k$ when $r \leq N$.

The idea of the selective DFT attacks is to select a proper $q(x)$ such that only certain spectral points are passed through the filter, and the others are blocked or are nulled to zero. This idea is also facilitated a new method to compute DFT without the need of the entire sequence.

2.3 The Complexity for Computing DFT

From Property 2, the complexity for computing each A_k is the cost for evaluating polynomial $a(x)$ at α^{-k}. In order to evaluate the complexity of computing DFT, we need some results on the computation in a finite field $GF(2^n)$ [5][6]. .

Lemma 1 *(a) The multiplication or division of two polynomials of degree n has the same complexity which is $O(\eta(n))$ exclusive-or operations where*

$$\eta(n) = n \log n \log\log n.$$

(b) The complexity for solving a system of m linear equations over $GF(2^n)$ is given by

$$O(m^{2.37}\eta(n))$$

exclusive-or operations ([3]).

(c) Using the Berlekamp algorithm to compute the minimal polynomial of a binary sequence with length m is

$$O(m \log(m))$$

exclusive-or operations.

According to Lemma 1, we have the following property.

Property 6 *The complexity for computing A_k using the definition to evaluate $a(x)$ at $\beta = \alpha^{-k}$ is equal to the sum of the following two steps*

(a) The complexity for computing β is $O(\log(k)\eta(n))$ Xor operations.

(b) The complexity for evaluate $a(x)$ at β is $O(deg(a(x))\eta(n))$ Xor operations.

Furthermore, the total complexity is

$$O((\log(k)\eta(n) + deg(a(x))\eta(n)) \approx O(N/2\eta(n)) Xor\ operations,$$

since the degree of $a(x)$ is averagely about $N/2$ and $\log(k) < deg(a(x)) \approx N/2$.

3 A Fast Algorithm for Computing DFT

From the definition of the DFT, computing the DFT of a binary sequence $\{a_t\}$ with period N needs an entire period of the sequence. In the following, we present a fast algorithm for computing DFT of the sequence which only needs L consecutive bits of the sequence. This method makes the use of the selective DFT attack method and the Berlekamp algorithm [1] [10]. The idea is to apply a filter $q(x)$ to the sequence which allows one nonzero spectral point, say A_k, together with its conjugates $A_k^{2^i}$ to pass, and the others are blocked. This can be done by Property 5.

Algorithm 1 The fast algorithm of computing DFT (FDFT)

Input: $(a_0, a_1, \cdots, a_{J-1}), J \leq N$.
Output: $(A_0, A_1, \cdots, A_{N-1})$, the DFT of the sequence.

Initial Process: Applying the Berlekamp-Massey algorithm to $(a_0, \cdots, a_{J-1})$, we get $p(x)$. Let L be the degree of $p(x)$. If $2L \leq J$, then $p(x)$ is the minimal polynomial of $\{a_t\}$. In this case, we continue the following process. Otherwise, return a failure, i.e., the DFT in this case is not computable.

Selection of Parameters

1. Find n such that $N \mid 2^n - 1$.
2. Pick a primitive polynomial of degree n, say $d(x)$, and define the finite field $GF(2^n)$ by $d(x) = x^n + \sum_{i=0}^{n-1} d_i x^i, d_i \in \{0, 1\}$, and let β be a root of $d(x)$ in $GF(2^n)$.

3. Generate an m-sequence $\{b_t\}$ using $f(x)$ with the initial state $b_t = Tr(\beta^t), t = 0, 1, \cdots, n-1$ where $Tr(x) = x + x^2 + \cdots, +x^{2^{n-1}}$, and $b_{n+t} = d_0 b_t + d_1 b_{t+1} + \cdots + d_{n-1} b_{t+n-1}, t \geq 0$.

4. Let α be an element in $GF(2^n)$ with order N.

Procedure

Step 1 Compute the coset leader set I using the method in Table 2.1.

Step 2 For each $k \in I$, compute $p(\alpha^k)$. If $p(\alpha^k) \neq 0$, then set $A_k = 0$ (here we use Property 3). Otherwise compute A_k using the selective DFT algorithm given below.

Step 3 Selective DFT Procedure. The following process is to filter out all the other nonzero spectra except for A_k and its conjugates. In other words, we generate the time convolution sequence $\{v_t\}$ from $q(x)$ and the input sequence, and then compute its spectral.

Procedure for A_k:

1. *Compute the coset size.* Compute m such that m is the smallest integer such that $k \equiv k2^m \,(\mathrm{mod} N)$. (m is the size of the coset $C_k = \{k, 2k, \cdots, k2^{m-1}\}$ modulo N.)
2. *Compute the minimal polynomial of k-decimation sequence of the m-sequence.* If $m = n$, compute $c_t = b_{tk}, t = 0, 1, \cdots, 2m-1$, k-decimation of $\{b_t\}$. If $m < n$, compute $c_t = Tr_m(\alpha^{kt}), t = 0, 1, \cdots, 2m-1$ where $Tr_m(x) = x + x^2 + \cdots + x^{2^{m-1}}$. Applying the BM algorithm to $(c_0, c_1, \cdots, c_{2m-1})$. We get $p_k(x)$, the minimal polynomial of this decimation sequence, and set an $m \times m$ matrix M using $\{c_t\}$ as follows
$$M = \begin{pmatrix} c_0 & c_1 & c_2 & \cdots & c_{m-1} \\ c_1 & c_2 & c_3 & \cdots & c_m \\ c_2 & c_3 & c_4 & \cdots & c_{m+1} \\ \vdots & & & & \\ c_{m-1} & c_m & c_{m+1} & \cdots & c_{2m-2} \end{pmatrix}$$
This matrix is called a circulant matrix generated by $\{c_t\}$.
3. *Construct a selective filter.* Set $q(x) = \frac{p(x)}{p_k(x)} = \sum_{i=0}^{l-1} q_i x^i$ where $l = L - m$.

4. *Compute the time convolution of $q(x)$ and $\{a_t\}$ and its spectra.* Compute the time convolution: $v_t, t = 0, 1, \cdots, m-1$ by applying $q(x)$ to $(a_0, \cdots, a_{L-1})$:

$$v_t = \sum_{i=0}^{l-1} q_i a_{i+t}, t = 0, 1, \cdots, m-1. \tag{9}$$

   Solve the following system of m linear equations in m variables $(x_0, \cdots, x_{m-1})$:

$$M \begin{pmatrix} x_0 \\ x_1 \\ \vdots \\ x_{m-1} \end{pmatrix} = \begin{pmatrix} v_0 \\ v_1 \\ \vdots \\ v_{m-1} \end{pmatrix}$$

5. Compute $V = \sum_{i=0}^{m-1} x_i \alpha^{ik}$ and $T = q(\alpha^k)^{-1}$. Then compute $A_k = VT$.

Step 4 Return $(A_0, A_1, \cdots, A_{N-1})$.

We use an example to demonstrate the procedure.

Example 3 Let $\{a_t\} = (100010010101010)$, a binary sequence of period 15. Find the DFT of $\{a_t\}$. Since we know the entire period of the sequence, we could use the definition of DFT to compute the DFT of the sequence in this case. However, in order to demonstrate the procedure, the DFT is computed using the above procedure.

Initial process: Applying the Berlekamp-Massey algorithm to $(a_0, \cdots, a_{14})$, we get

$$p(x) = x^{10} + x^8 + x^5 + ^4 + x^2 + x + 1,$$

the minimal polynomial of $\{a_t\}$ and $L = 10$. Since we have an entire period of the sequence, thus Algorithm 1 is applicable.

Selection of Parameters

1. Find $n = 4$, since $N = 15$.

2. Choose a primitive polynomial of degree 4 as $f(x) = x^4 + x + 1$, and the finite field $GF(2^4)$ is defined by $f(x)$. Let α be a root of $f(x)$ in $GF(2^4)$. Since $N = 15$, we can choose α in this way.

3. Generate an m-sequence $\{b_t\}$ using $f(x)$ with the initial state $b_t = Tr(\alpha^t), t = 0, 1, 2, 3$ where $Tr(x) = x + x^2 + x^4 + x^8$:

$$b_0 = 0, b_1 = Tr(\beta) = 0, b_2 = Tr(\beta^2) = 0, b_3 = Tr(\beta^3) = 1.$$

Thus

$$\{b_t\} = (000100110101111)$$

Procedure

Step 1. Compute the coset leader set $I = \{0, 1, 3, 5, 7\}$.

Step 2. For each $k \in I$, note that $A_0 = 0$. Compute $p(\alpha^k)$ for $k \in I \Longrightarrow p(\alpha^k) = 0, k \in I_0 = \{1, 3, 5\}$.

Step 3. *Selective DFT algorithm:* Procedure for computing $A_k, k \in I_0$:

Case $k = 1$**.**

1. *Coset size:* Compute m such that m is the smallest integer such that $1 \equiv 2^m \,(\text{mod} 15)$ $\Longrightarrow m = 4$.
2. *1-decimation of the m-sequence.* Here $p_1(x) = f(x)$, so $c_t = b_t$. Set an 4×4 matrix M

$$M = \begin{pmatrix} b_0 & b_1 & b_2 & b_3 \\ b_1 & b_2 & b_3 & b_4 \\ b_2 & b_3 & b_4 & b_5 \\ b_3 & b_4 & b_5 & b_6 \end{pmatrix} = \begin{pmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 \end{pmatrix}$$

3. *Construct a selective filter.* Set $q(x) = \frac{p(x)}{p_1(x)} = x^6 + x^4 + x^3 + x^2 + 1$ where $l = L - m = 6$.
4. *Compute the time convolution:* $v_t, t = 0, 1, 2, 3$ by applying $q(x)$ to $\{a_t\}$:

$$v_t = a_t + a_{t+2} + a_{t+3} + a_{t+4} + a_{t+6}, t = 0, 1, 2, 3 \Longrightarrow (v_0, v_1, v_2, v_3) = (0, 0, 1, 0).$$

Solve the following system of 4 linear equations in 4 variables (x_0, x_1, x_2, x_3):

$$M \begin{pmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} v_0 \\ v_1 \\ v_2 \\ v_3 \end{pmatrix} \Longrightarrow \begin{pmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \end{pmatrix} \Longrightarrow \begin{pmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \end{pmatrix}.$$

5. Compute $V = \sum_{i=0}^{3} x_i \alpha^i = \alpha$ and $q(\alpha) = \alpha \Longrightarrow T = q(\alpha)^{-1} = \alpha^{14}$. Set $A_1 = VT = \alpha\alpha^{14} = 1$.

Case $k = 3$.

1. *Coset size.* Compute m such that m is the smallest integer such that $3 \equiv 3\times 2^m \pmod{15}$ $\Longrightarrow m = 4$.

2. *3-decimation of the m-sequence.* Since $m = 4$, let $c_t = b_{3t}, t = 0, 1, \cdots, 7$, then $\{c_t\}$ has period 4

$$(c_0, c_1, \cdots,) = (0, 1, 1, 1, 1)$$

   with period 5. Applying the BM to the above sequence, we get $p_3(x) = x^4+x^3+x^2+x+1$. Set an 4×4 matrix M

$$M = \begin{pmatrix} c_0 & c_1 & c_2 & c_3 \\ c_1 & c_2 & c_3 & c_4 \\ c_2 & c_3 & c_4 & c_0 \\ c_3 & c_4 & c_0 & c_1 \end{pmatrix} = \begin{pmatrix} 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 \\ 1 & 1 & 0 & 1 \end{pmatrix}$$

3. *Selective filter:* Set $q(x) = \frac{p(x)}{p_3(x)} = x^6 + x^5 + x^4 + x^3 + 1$ where $l = L - m = 6$.

4. *Compute the time convolution:* $v_t, t = 0, 1, 2, 3$ by applying $q(x)$ to $\{a_t\}$:

$$v_t = a_t + a_{t+3} + a_{t+4} + a_{t+5} + a_{t+6}, t = 0, 1, 2, 3 \Longrightarrow (v_0, v_1, v_2, v_3) = (0, 0, 1, 0).$$

   Solve the following system of 4 linear equations in 4 variables (x_0, x_1, x_2, x_3):

$$\begin{pmatrix} 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 \\ 1 & 1 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \end{pmatrix} \Longrightarrow \begin{pmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \\ 0 \\ 1 \end{pmatrix}.$$

5. Compute $V = \sum_{i=0}^{3} x_i \alpha^{3i} = \alpha^3 + \alpha^9 = \alpha$ and $q(\alpha^3) = \alpha^{13} \Longrightarrow T = q(\alpha^3)^{-1} = \alpha^2$. Set $A_3 = VT = \alpha\alpha^2 = \alpha^3$.

Case $k = 5$**.** $m = 2$, $(c_0, c_1, c_2) = (0, 1, 1)$, $p_5(x) = x^2 + x + 1 \Longrightarrow q(x) = x^8 + x^7 + x^6 + x^4 + 1$ where $l = L - m = 8$, $(v_0, v_1) = (1, 0)$,

$$M = \begin{pmatrix} c_0 & c_1 \\ c_1 & c_2 \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ 1 & 1 \end{pmatrix} \Longrightarrow (x_0, x_1) = (1, 1)$$

Thus $V = x_0 + x_1\alpha^5 = 1 + \alpha^5 = \alpha^{10}$ and $q(\alpha^5) = \alpha^{10} \Longrightarrow T = q(\alpha^5)^{-1} = \alpha^5$. Compute $A_5 = VT = \alpha^{10}\alpha^5 = 1$.

Thus we get

$$(A_0, A_1, \cdots, A_{14}) = (0, 1, 1, \alpha^3, 1, 1, \alpha^6, 0, 1, \alpha^9, 1, 0, \alpha^{12}, 0, 0)$$

Property 7 *The complexity of Algorithm 1 is the sum of the following three dominant factors.*

(a) Compute $p(x)$ by applying the Berlekamp algorithm, the complexity $O(L\log(L))$ Xor operations.

(b) Compute $p(\alpha^k)$ with the complexity $O(L\eta(n))$ Xor operations for each $k \in I$.

(c) Compute $q(x) = p(x)/p_k(x)$ for $k \in I_0$ with the complexity $Q(\eta(L))$ Xor operations.

Comparing with Property 6, the saving of Algorithm 1 is given by (b) if $L < deg(a(x)) \leq N$ because the degree of $a(x)$ could be as large as N. Furthermore, Algorithm 1 does not need the entire sequence.

4 Selective DFT Attacks

In this section, we will fist use a filter generator to demonstrate the selective DFT attacks. Then we explain a scenario for applications for combinatorial generators as well as hash functions and block ciphers.

4.1 Reference pair model for the selective DFT attack

Let $t(x)$ be the characteristic polynomial of the LFSR, which is primitive and $f(\mathbf{x})$ where $\mathbf{x} = (x_0, x_1, \cdots, x_{n-1})$ be a boolean function in n variables. Then any output of the filtering generator can be represented by

$$u_t = f(b_t, b_{t+1}, \cdots, b_{t+n-1}), t = 0, 1, \cdots,$$

where $\{b_t\}$ be an m-sequence generated by the LFSR, regardless an actual boolean function used in the filtering generator. We may write

$$u_t = f(\mathbf{b}_t), \text{where } \mathbf{b}_t = (b_t, b_{t+1}, \cdots, b_{t+n-1}), t = 0, 1, \cdots .$$

The problem arisen here is that given a segment of the output sequence $\{s_t\}$, find the initial state of the m-sequence, i.e., $\mathbf{b}_0 = (b_0, \cdots, b_{n-1})$.

We also use $t(x)$ to define $\mathbb{F}_{2^n}$ and assume that α is a root of $t(x)$ in $\mathbb{F}_{2^n}$. In the following, all DFTs are computed using α where $N = 2^n - 1$. Let $\{a_t\}$ be an m-sequence generated by the LFSR with an initial state $a_t = Tr(\alpha^t), t = 0, 1, \cdots, n-1$ where $Tr(x) = x + x^2 + \cdots + x^{2^{n-2}}$, the trace function, and

$$s_t = f(\mathbf{a}_t), \text{where } \mathbf{a}_t = (a_t, a_{t+1}, \cdots, a_{t+n-1}), t = 0, 1, \cdots . \tag{10}$$

We call $(\mathbf{a}, \mathbf{s})$ a *reference pair* of the generator. Since both m-sequences $\{a_t\}$ and $\{b_t\}$ are generated by the same LFSR, i.e., the same primitive polynomial $t(x)$, they are shift equivalent. So we may write

$$b_t = a_{t+\tau}, t = 0, 1, \cdots, \tag{11}$$

Moreover, both filtering sequences $\{s_t\}$ and $\{u_t\}$ are also shift equivalent, which has the same shift as that of $\{a_t\}$ to $\{b_t\}$, i.e.,

$$u_t = s_{t+\tau}, t = 0, 1, \cdots . \tag{12}$$

Let the DFTs of $\{s_t\}$ and $\{u_t\}$ be $\{S_k\}$ and $\{U_k\}$ respectively . According to the definition of the DFT, we have

$$U_k = \alpha^{\tau k} S_k, 0 \le k < N \tag{13}$$

where τ is the shift of $\{a_t\}$ which produces $\{b_t\}$ in (11).

It is a very interesting phenomenon that the exact information on the time shift occurs at the every spectral point. Thus we only need to find nonzero spectral U_k and S_k for only one k such that $\gcd(k, 2^n - 1) = 1$. Usually, we pick the smallest k such that $\gcd(k, 2^n - 1) = 1$ and α^k is a root of the minimal polynomial of $\mathbf{s}$. For such k, from (13), we have

$$\beta = (U_k S_k^{-1})^{k^{-1}}, \ \beta = \alpha^\tau .$$

Thus the initial state of the LFSR which generates $\{s_t\}$ is given by

$$b_t = Tr(\beta \alpha^t), t = 0, 1, \cdots, n-1.$$

Now finding the initial state of $\mathbf{b}$ is converted to find S_k and U_k for a particular k. The idea of the selective DFT attack is to use the reference sequences $(\mathbf{a}, \mathbf{s})$ for computing S_k using the *selective filter*

$$q(x) = \frac{p(x)}{p_k(x)} = \sum_{i=0}^{l} c_i x^i, c_i \in \mathbb{F}_2 = \{0, 1\}$$

where $p(x)$ and $p_k(x)$ are their respective minimal polynomials of $\mathbf{s}$ and an k-decimation of $\mathbf{a}$: $a_0, a_k, a_{2k}, \cdots$.

Let $\{v_t\}$ and $\{w_t\}$ be the discrepancy sequences associated with $q(x)$ and $\mathbf{s}$ and $q(x)$ and $\mathbf{u}$ respectively, we have

$$V_k = q(\alpha^k)S_k \neq 0 \Longrightarrow S_k = V_k[q(\alpha^k)]^{-1}$$

$$W_k = q(\alpha^k)U_k \neq 0 \Longrightarrow U_k = W_k[q(\alpha^k)]^{-1}$$

where $\{V_k\}$ and $\{W_k\}$ are the DFTs of $\{v_t\}$ and $\{w_t\}$ respectively. Thus, S_k or U_k can be computed in terms of V_k or W_k and $q(\alpha^k)$ where $q(\alpha^k)$ is known. The computation of $q(x)$ and S_k are referred to as *off-line computation* or *pre-computation*, while the computation of W_k, therefore U_k, is called *on-line computation* since it involves the key, i.e., the initial state used for producing the sequence $\{u_t\}$. The relationship among those sequences is shown in Figure 5.

Note that V_k and W_k can be computed in terms of $(v_0, \cdots, v_{n-1})$ and $(w_0, \cdots, w_{n-1})$ by solving the following linear equations in $GF(2^n)$ respectively:

$$v_i = Tr(V_k\alpha^{ki}), i = 0, 1, \cdots, n-1 \text{ linear equations in unknown } V_k \tag{14}$$

$$w_i = Tr(W_k\alpha^{ki}), i = 0, 1, \cdots, n-1 \text{ linear equations in unknown } W_k \tag{15}$$

However, those systems of the linear equations over the extension field $GF(2^n)$ can be converted to the systems of linear equations over $GF(2)$.

4.2 Example

In the following, we use an example to explain this method in details.

Example 4 Let a filter generator with the following parameters.

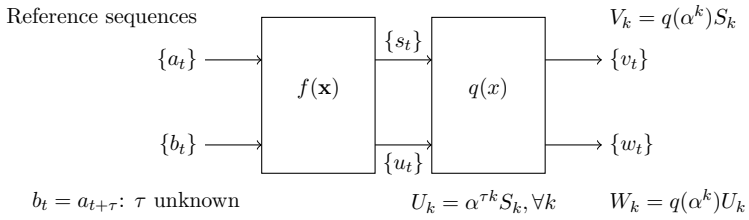

Figure 5: Reference sequence model for the selective DFT attack where $\{S_k\}$, $\{U_k\}$, $\{V_k\}$, and $\{W_k\}$ are their respective DFTs of $\{s_t\}$, $\{u_t\}$, $\{v_t\}$ and $\{w_t\}$, and $q(x) = p(x)/p_k(x)$ is the selective filter for $q(\alpha^k) \neq 0$ and $q(\alpha^t) = 0$ for all $t \not\equiv k2^j \bmod 2^n - 1$.

LFSR	$t(x) = x^4 + x + 1$
Tap positions	$(d_0, d_1, d_2, d_3) = (0, 1, 2, 3)$
Filtering function	$f(x_0, x_1, x_2, x_3) = x_0 + x_3 + x_0x_3 + x_1x_3 + x_0x_1x_3 + x_0x_1x_2x_3$

Let $\{b_t\}$ be an output sequence of the LFSR and $\{u_t\}$ be the output of the filter function f whose elements are given by

$$u_t = f(\mathbf{b}_t), \mathbf{b}_t = (b_t, b_{t+1}, b_{t+2}, b_{t+3}), t = 0, 1, \cdots, 14.$$

Assume that 10 bits $\mathbf{u}^{10} = 1011101000$ are known to an attacker. Find the initial state of the LFSR which generates $\mathbf{u}^{10}$.

Solution.

Off-line computation

1. Compute the reference pair $(\mathbf{a}, \mathbf{s})$ as follows: Let $\{a_t\}$ be generated by $f(x)$ with the initial state $a_t = Tr(\alpha^t), t = 0, 1, 2, 3 \Longrightarrow (a_0, a_1, a_2, a_3) = (0, 0, 0, 1)$. We generate

$$s_t = g(\mathbf{b}_t), \mathbf{a}_t = (a_t, a_{t+1}, a_{t+2}, a_{t+3}), t = 0, 1, \cdots, 14 \Longrightarrow \{s_t\} = 100110111010001.$$

2. Computing the selective filter $q(x)$:

   (a) Computing $p(x)$, the minimal polynomial of $\mathbf{s}$: Run the BM algorithm to $\{s_t\}$, we get the minimal polynomial $p(x) = x^{10} + x^9 + x^8 + x^6 + x^5 + x^2 + 1$ which generates $\{s_t\}$ and the linear span $L = 10$.

(b) Note that there are only 1 and 7 are coprime with 15 (not including the indexes 2^i and 7×2^i mod 15), we check: $p(\alpha) \neq 0$ and $p(\alpha^7) = 0$. Thus we select $k = 7$.

(c) Compute $c_t = a_{7t}, t = 0, 1, \cdots, 7 \Longrightarrow (c_0, \cdots, c_7) = (01111010)$. Apply the BM algorithm to $\{c_t\}$, we obtain $p_7(x) = x^4 + x^3 + 1$ which is the minimal polynomial of $\{c_t\}$. Set the selective filter: $q(x) = p(x)/p_7(x) = x^6 + x^4 + x^3 + x^2 + 1$ and $l = L - m = 6$, and compute $q(\alpha^7) = \alpha^5 \Longrightarrow T = q(\alpha^7)^{-1} = \alpha^{10}$.

3. Compute S_7 using the selective DFT algorithm in Algorithm 1.

Selective DFT procedure

(a) Set M to be a matrix defined below:

$$M = \begin{pmatrix} c_0 & c_1 & c_2 & c_3 \\ c_1 & c_2 & c_3 & c_4 \\ c_2 & c_3 & c_4 & c_5 \\ c_3 & c_4 & c_5 & c_6 \end{pmatrix}$$

(this is referred to as the *circulant matrix* generated by $\{c_t\}$). Substituting the values of $c_t, t = 0, 1, \cdots, 7$, we have

$$M = \begin{pmatrix} 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 \\ 1 & 1 & 0 & 1 \end{pmatrix}.$$

(b) Computing the time convolution of $q(x)$ and $\{s_t\}$: $v_t = s_t + s_{t+2} + s_{t+3} + s_{t+4} + s_{t+6}, t = 0, 1, 2, 3 \Longrightarrow$

$$(v_0, v_1, v_2, v_3) = (0, 1, 1, 1).$$

Solve the following system of the linear equations in unknowns $\mathbf{x} = (x_0, x_1, x_2, x_3)$:

$$M\mathbf{x}^T = \mathbf{v}^T \Longrightarrow \begin{pmatrix} 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 \\ 1 & 1 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \\ 1 \\ 1 \end{pmatrix} \Longrightarrow \begin{pmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix}.$$

Compute $V = \sum_{i=0}^{3} x_i \alpha^{7i} = 1$. Set

$$S_7 = Vq(\alpha^7)^{-1} = \alpha^{10}.$$

On-line computation:

1. With the same $\{c_t\}$, M, and $q(x)$ as those for computing S_7, using the selective DFT procedure to compute U_7.

   (a) M is the same as for S_7.

   (b) Compute the time convolution of $q(x)$ and $\{u_t\}$:

   $$w_t = u_t + u_{t+2} + u_{t+3} + u_{t+4} + u_{t+6}, t = 0, 1, 2, 3 \Longrightarrow (w_0, w_1, w_2, w_3) = (1, 0, 1, 0).$$

   Solve for the unknowns (x_0, x_1, x_2, x_3) in the system of the linear equations: $M\mathbf{x}^T = \mathbf{w}^T$:

   $$\begin{pmatrix} 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 \\ 1 & 1 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \\ 1 \\ 0 \end{pmatrix} \Longrightarrow \begin{pmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 1 \end{pmatrix}.$$

   Compute $W = \sum_{i=0}^{3} x_i \alpha^{7i} = 1 + \alpha^6 = \alpha^{13}$. Set

   $$U_7 = Wq(\alpha^7)^{-1} = \alpha^{13}\alpha^{10} = \alpha^8.$$

2. Compute $\alpha^\tau = (S_7 U_7^{-1})^{13} = \alpha^4$ where $7^{-1} = 13 \mod 15$.

3. Thus $b_t = a_{t+4} \Longrightarrow (b_0, b_1, b_2, b_3) = (0, 0, 1, 1)$.

Thus the filtering generator generates $\mathbf{u}^{10}$ when the initial state of the LFSR is $(b_0, b_1, b_2, b_3) = (0, 0, 1, 1)$.

We show the evolution of the nonzero spectral points involved in the selective DFT attack in Figure 6. For simplicity, the magnitude of the nonzero spectral is mapped to a positive constant. The indexes listed are only those in the set $\{0, 1, 3, 5, 7\}$, since the spectral points at the indexes with $k2^i \mod 15$ are zero if and only if the spectral points is zero at k. For example, for the m-sequence $\mathbf{a}$, A_1, A_2, A_4, A_8 are nonzero since A_1 is nonzero. Thus, both

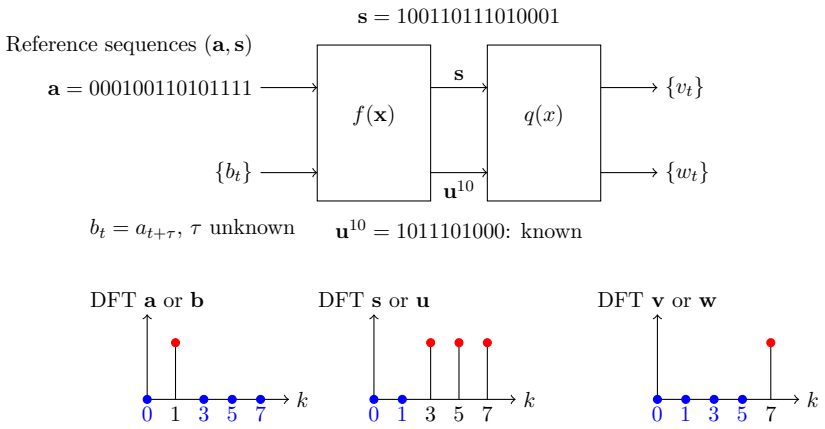

Figure 6: The reference sequences and the evolution of the DFT spectral points involved in the example.

m-sequences **a** and **b** have four nonzero spectral points, which is the degree of the LFSR. The filtering sequence **s** or **u** has ten nonzero spectral points, which have the representative points at 3, 5, and 7. After the selective filter $q(x)$, the output sequence **v** or **w** has four nonzero spectral points at 7, 14, 13, 11 where 7 is a representative. The number of nonzero spectral points of **v** or **w** is the same as the m-sequence **a** or **b**. The effect of the filter $q(x)$ is to null some spectral points and only to allow the nonzero spectral points which are interested to the attacker to pass, so this is why it is called the *selective* DFT attack.

4.3 An Algorithm of the selective DFT attack using the reference pair

We now present a selective DFT attack algorithm for a filtering generator. Then we explain how to apply to the other cases.

Algorithm 2 *(*Selective DFT attack algorithm*)*

Input: $t(x)$ is a primitive characteristic polynomial of degree n for the LFSR, $\{b_t\}$, the output of the LFSR, $f(\mathbf{x}), \mathbf{x} = x_0, \cdots, x_{n-1})$ is a boolean function in n variables, and $u_0, u_1, \cdots, u_{J-1}$ is the J consecutive bits of the filtering generator where $u_t = f(\mathbf{b}_t), \mathbf{b}_t = (b_t, b_{t+1}, \cdots, b_{t+n-1}), t = 0, 1, \cdots, J-1$. The finite field $GF(2^n)$ is defined by $t(x)$ and α in the DFT computation, a root of $t(x)$ in $GF(2^n)$.

Output: $(b_0, \cdots, b_{n-1})$, the initial state of LFSR.

Procedure

Off-line computation.

1. Compute the reference pair $(\mathbf{a}, \mathbf{s})$: Let $\{a_t\}$ be generated by $t(x)$ with the initial state $a_t = Tr(\alpha^t), t = 0, 1, \cdots, n-1$ and generate

$$s_t = f(\mathbf{a}_t), \mathbf{a}_t = (a_t, a_{t+1}, \cdots, a_{t+n-1}), t = 0, 1, \cdots .$$

2. Compute the selective filter $q(x)$:

   (a) Applying the BM algorithm to $\{s_t\}$, we get $p(x)$ which is the minimal polynomial of $\{s_t\}$ and the linear span L. If $J \geq L$, then continue the following process. Otherwise, return a *failure*.

   (b) Find the first k such that $\gcd(k, 2^n - 1) = 1$ and $p(\alpha^k) = 0$, then compute k^{-1} mod $2^n - 1$.

   (c) For this k, compute $c_t = a_{tk}, t = 0, 1, \cdots, 2n-1$. Applying the BM algorithm to $\{c_t\}$, we get $p_k(x)$, set $q(x) = p(x)/p_k(x) = \sum_{i=0}^{l} c_i x^i$.

3. Compute S_k by the selective DFT algorithm in Algorithm 1.

   Selective DFT procedure

(a) Set

$$M = \begin{pmatrix} c_0 & c_1 & c_2 & \cdots & c_{n-1} \\ c_1 & c_2 & c_3 & \cdots & c_n \\ c_2 & c_3 & c_4 & \cdots & c_{n+1} \\ \vdots & & & & \\ c_{n-1} & c_n & c_{n+1} & \cdots & c_{2n-2} \end{pmatrix}$$

This matrix is called a *circulant matrix* generated by $\{c_t\}$.

(b) Compute the time convolution of $q(x)$ and $\{s_t\}$:

$$v_t = \sum_{i=0}^{l} q_i s_{i+t}, t = 0, 1, \cdots, n-1. \tag{16}$$

Solve the following system of m linear equations in n variables $(x_0, \cdots, x_{n-1})$:

$$M \begin{pmatrix} x_0 \\ x_1 \\ \vdots \\ x_{n-1} \end{pmatrix} = \begin{pmatrix} v_0 \\ v_1 \\ \vdots \\ v_{n-1} \end{pmatrix}$$

Compute $V = \sum_{i=0}^{n-1} x_i \alpha^{ik}$ and $S_k = V q(\alpha^k)^{-1}$.

On-line computation

1. With the same $\{c_t\}$, M, and $q(x)$ as those for computing S_k, compute U_k.

   (a) M is the same for S_k.

   (b) Compute the time convolution of $q(x)$ and $\{u_t\}$:

$$w_t = \sum_{i=0}^{l-1} q_i u_{i+t}, t = 0, 1, \cdots, n-1.$$

Solve for the unknowns $\mathbf{x} = (x_0, \cdots, x_{n-1})$ in the system of the linear equations:

$$M\mathbf{x}^T = \mathbf{w}^T.$$

Set $W = \sum_{i=0}^{n-1} x_i \alpha^{ki}$ and

$$U_k = W q(\alpha^k)^{-1}$$

2. Compute $\beta = (S_k U_k^{-1})^{k^{-1}}$.

3. Return the initial state $b_t = Tr(\beta \alpha^t), t = 0, 1, \cdots, n-1$.

It is worth to point it out that the shift-equivalent relationships are well preserved at the output sequences, which makes this attack available. In the following, we summarize those relationships in Table 5.

Table 5: Observations of selective DFT attacks on filtering generators

LFSR of degree n	Filtering sequences
primitive polynomial $t(x)$	$f(\mathbf{x}), \mathbf{x} = (x_0, \cdots, x_{n-1})$, boolean in n variables
$\{a_t\}$ (selected by attacker) and $\{b_t\}$, unknown to the attacker, are generated by the LFSR	$s_t = f(\mathbf{a}_t)$, computed by the attacker, $\mathbf{a}_t = (a_t, a_{t+1}, \cdots, a_{t+n-1})$, $u_t = f(\mathbf{b}_t)$, $\mathbf{b}_t = (b_t, b_{t+1}, \cdots, b_{t+n-1})$, J consecutive bits are known
Then $b_t = a_{t+\tau}$	$u_t = s_{t+\tau}$, and $U_k = \alpha^{k\tau} S_k$
The shift τ or equivalently, the initial state of the LFSR which generates $\{b_t\}$ can be found by computing only one spectral point in each sequence provided that $J > L$ consecutive bits of $\{u_t\}$ is known to the attacker where L is the linear span of any output sequence of the generator (they all have the same linear span).	

If the linear span is high, the selective DFT method cannot be directly used. However, it can use the method such as multiplying another sequence such that the product sequence has reduced linear span.

The above scenario can be applied to a block cipher or a MAC for finding the key, which we outline in Table 6.

The degree n of the LFSR can be chosen such that the linear span of $\{s_t\}$ is computable. In general $E_K(x)$ will make the sequence $\{s_t\}$ to reaches the maximum linear span. In this case, it may happen the following properties about the output vector sequences in Table 6.

Table 6: Applying selective DFT attacks to block cipher $E_K(x)$ (Bo Zhu, 2011)

K, m bits, find K. LFSR of degree $n < m$	$E_K(x) : \mathbb{F}_2^m \longrightarrow \mathbb{F}_2^m$ Apply $E_K(x)$ as a filtering function
primitive polynomial $t(x)$	$E_K(\mathbf{x}), \mathbf{x} = (x_0, \cdots, x_{n-1})$, boolean in n variables
$\{a_t\}$ and $\{b_t\}$, which can selected according to some structure, are generated by the LFSR	$\mathbf{s}_t = E_K(\mathbf{a}_t, \mathbf{0})$, $\mathbf{a}_t = (a_t, a_{t+1}, \cdots, a_{t+n-1})$, $\mathbf{u}_t = E_K(\mathbf{b}_t, \mathbf{0})$, $\mathbf{b}_t = (b_t, b_{t+1}, \cdots, b_{t+n-1})$, where $\mathbf{0}$ is the zero vector of dimensional $m-n$, both $\mathbf{s}_t$ and $\mathbf{u}_t$ are m bits vectors. Those can be obtained by feeding $\mathbf{a}_t$ or $\mathbf{b}_t$ as plaintext by padding $m-n$ zero bits.
Then $b_t = a_{t+\tau}$	$u_t = s_{t+\tau}$, and $U_k = \alpha^{k\tau} S_k$ where $\{s_t\}$ and $\{u_t\}$ represent the corresponding component sequences in their respective output vector sequences.

1. Period for each component sequence is $2^n - 1$.

2. In general, the linear span of each component sequence may reach the maximum $2^n - 1$.

3. How to recover K from the DFT $\{S_k\}$ and $\{U_k\}$, which contain rich information of K? In other words, we can consider that each component is a function of K.

5 Discussions

We have presented a fast computation procedure for DFT without requiring the full period of a sequence and introduced a simplified selective DFT attack using the reference model. We also demonstrated how this method could be used to attack hash functions and block ciphers, which is proposed by Bo Zhu recently. However, it is unsolved how to relate keys with the initial states of reference LFSRs introduced to attack the systems turned from hash functions and clock ciphers.

References

[1] E.R. Berlekamp, *Algebraic coding theory*, New York, McGraw-Hill, 1968.

[2] R.E. Blahut, *Theory and Practice of Error Control Codes*, Addison-Wesley, 1983.

[3] D. Coppersmith and S. Winograd, Matrix multiplication via arithmetic programing, *Journal of Symbolic Computation*, vol. 9, pp. 251-280, 1990.

[4] G. Gong, S. Rønjom, T. Helleseth, and H.G. Hu, Fast Discrete Fourier Spectra Attacks on Stream Ciphers, to appear in IEEE IT soon.

[5] Philip Hawkes and Gregory G. Rose, Rewriting variables: the complexity of fast algebraic attacks on stream ciphers, *Advances in Cryptology-Crypto'2004*, Lecture Notes in Computer Science, No. 3152, pp. 390-406, Springer-Verlag, 2004.

[6] E. Kaltofen and V. Shoup, Subquadratic-time factoring of polynomials over finite fields, *Mathematics of Computation*, vol. 67, No. 223, July 1998, pp. 11791197, 1998.

[7] S.W. Golomb, *Shift Register Sequences,* Holden-Day, Inc., San Francisco, 1967, revised edition, Aegean Park Press, Laguna Hills, CA, (1982).

[8] S.W. Golomb and G. Gong, *Signal Design with Good Correlation: for Wireless Communications, Cryptography and Radar Applications*, Cambridge University Press, 2005.

[9] F.J. MacWilliams, N.J.A. Sloane, *The Theory of Error-Correcting Codes*, North-Holland Publishing Company, 1977.

[10] J. L. Massey, Shift-register synthesis and BCH decoding, *IEEE Trans. Information Theory*, Vol. 15, No. 1, pp. 122-127, January 1969.

[11] M.B. Pursley, *Introduction to Digital Communications*, Prentice Hall, 2005.

[12] S. Rønjom and T. Helleseth, A New Attack on the Filter Generator, *IEEE Transactions on Information Theory*, vol. 53, no. 5, pp. 1752-1758, 2007.

[13] S. Rønjom, G. Gong and T. Helleseth, On attacks on filtering generators using linear subspace structures, *Sequences, Subsequences, and Consequences*, *Lecture Notes in Computer Science*, S.W. Golomb *et al.* (Eds.), vol. 4893, pp. 204-217. Springer-Verlag, 2007.